Word Order Matters When You Increase Masking

Karim Lasri[ə,β] **Alessandro Lenci**[β] **Thierry Poibeau**[ə]
[ə]Lattice (École Normale Supérieure-PSL, CNRS, U. Sorbonne Nouvelle)
[β]University of Pisa
karim.lasri@ens.psl.eu
alessandro.lenci@unipi.it thierry.poibeau@ens.psl.eu

Abstract

Word order, an essential property of natural languages, is injected in Transformer-based neural language models using position encoding. However, recent experiments have shown that explicit position encoding is not always useful, since some models without such feature managed to achieve state-of-the art performance on some tasks. To understand better this phenomenon, we examine the effect of removing position encodings on the pre-training objective itself (i.e., masked language modelling), to test whether models can reconstruct position information from co-occurrences alone. We do so by controlling the amount of masked tokens in the input sentence, as a proxy to affect the importance of position information for the task. We find that the necessity of position information increases with the amount of masking, and that masked language models without position encodings are not able to reconstruct this information on the task. These findings point towards a direct relationship between the amount of masking and the ability of Transformers to capture order-sensitive aspects of language using position encoding.

1 Introduction

Transformer-based language models have become ubiquitous since they demonstrated improvements in most NLP downstream tasks (Devlin et al., 2019; Liu et al., 2019). A lot of ink has been spilled regarding the amount of linguistic structure that such models captured (Jawahar et al., 2019), pointing towards the acquisition of diverse linguistic abilities. As neural language models need to process information about the position of their input tokens to capture structural generalizations, a plethora of proposals to encode such information have been made (Press et al., 2021; He et al., 2020; Su et al., 2021; Chang et al., 2021; Chen et al., 2021; Chen, 2021). Recent work, however, questioned whether word order information is really useful for pre-trained models to solve downstream tasks (Sinha et al., 2021), showing that models could perform well when using only higher-order co-occurrence statistics. Other authors (Haviv et al., 2022) have shown that some transformers could reconstruct partly position information without it being explicitly injected. Examining performance on downstream tasks can show that the task simply does not require order information, or that the dataset used to test the model is too easy (Abdou et al., 2022), leading to indirect observations regarding a model's ability to reconstruct position information.

In turn, we choose to test the importance of position encodings for the pre-training task itself, masked language modeling, to get more direct evidence about whether and when position matters to language models. We do so under different amounts of masking, as intuitively, position information should be increasingly important when more tokens are missing from the context. Our experiments show that when masking only one token, the absence of position encoding has little effect on the model's performance. However, its importance increases with the number of masked tokens, forcing the model to leverage position information to perform better on its training objective. This finding should draw our attention towards choosing more carefully the amount of masking to train masked language models – a choice as important as the position encoding scheme itself.

2 Related work

A recent line of research investigated the importance of word order information during pre-training for models to solve downstream tasks, showing little variations when their input sentences are shuffled (Pham et al., 2021; Sinha et al., 2021; Hessel and Schofield, 2021). In a similar line of research, (Haviv et al., 2022) found that even in absence of position encoding, models were still able to reconstruct the latter when probed for tokens' absolute

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 1808 - 1815

position information in their intermediate layers. This finding in turn questioned the need for injecting explicitly position information in language models. (Abdou et al., 2022) also showed that shuffled models were still able to capture position information even when information about word order was removed after subword segmentation, likely because of the dependency between unigram occurrence probability and sentence length. Given all this work, it is surprising that the importance of explicit word order information in a neural language model still eludes us. In this study, we choose to investigate more carefully this phenomenon, and propose a methodology carefully designed to evaluate the importance of position encoding for the pre-training objective.

3 Experimental Setup

3.1 Methodology

In our experiments, the goal is to investigate the extent to which a transformer neural model requires explicit position encoding to perform well on the masked language modeling objective. We do so under different amounts of masking to examine how this parameter affects the need for explicit position encoding. We make use of two variants for each trained model, one in which we inject position information, and one deprived from explicit access to that information. To evaluate whether the trained model reconstructs its input sentences using position information, we compare its probability estimates q to two versions of the language's true probabilities p on the validation set. The first version, p_o, represents the probability of completions given the original, ordered input context. The second version, p_u, is the probability given unordered contexts. In the following sections, we explain how we perform this comparison to evaluate the extent to which explicit position encoding is required for the masked language modeling task.

3.2 Data

When using natural languages, it is hard to assess whether the model indeed relies on order information because it is not easy to design a dataset controlled to target specifically the usage of position information. In particular, as one does not have access to the true probability distribution of natural languages, it is hard to make clear predictions regarding how a model not using position information should behave. On the other hand, artificial

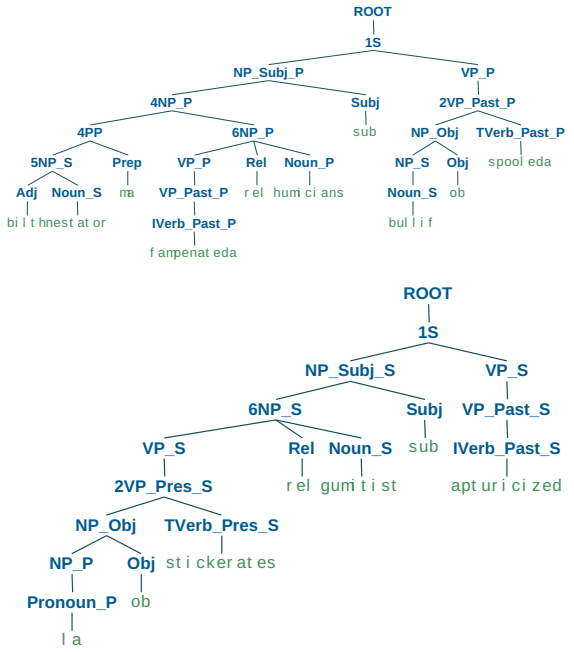

Figure 1: Examples of sentences found in the artificial language used for our analysis.

languages obtained from a generative procedure that is known *a priori* make it possible to get tight estimates of their true probability distribution, both with, and without access to position information. The use of artificial languages has sparked interest over the past years, as a proxy to test targeted properties of neural models in controlled settings (White and Cotterell, 2021; Wang and Eisner, 2016). In our experiments, we make use of data released by White and Cotterell (2021).The dataset consists of sentences generated from an artificial grammar, using a CFG such that all production rules have fixed probabilities.[1] This design makes it possible to evaluate the true probability of completions given masked input sentences, as a comparison point to the model's observed behavior. We display examples of generated trees in Fig. 1.[2]

3.3 Estimating the true probability distribution of the task

We exploit our direct access to the generative procedure which produces our input sentences to estimate the true probability distribution of the masked language modeling task. We do so by assuming that

[1]The artificial language features certain constraints present in natural languages such as morphological agreement relations.

[2]In our experiments, we used the unaffected artificial language (Grammar 000000) released by White and Cotterell (2021). The original code can be found at https://github.com/rycolab/artificial-languages.

the context is either ordered or not. Specifically, we generate sequences recursively using the artificial language's production rules, until the probability sum of fully expanded sentences[3] reaches a certain coverage.[4] We then iterate over these sentences to mask words at each position and aggregate completions for sequences that share the same unmasked context in the Masked Language Modelling setting. We thus obtain a probability distribution of completions Y given (ordered) masked contexts X_o, which we write $\{p_o(y|x) \mid x, y \in X_o \times Y_o\}$.[5]

We also compute a second version of the probability distribution that assumes no ordering of the context, aggregating completions for unmasked sequences whose unordered masked context is the same in X_u, obtaining $\{p_u(y|x) \mid x, y \in X_u \times Y_u\}$. To get the probability for unordered contexts, we simply group input sequences by sorting their elements alphabetically to remove order information and sum their probabilities for each unordered context. As we only use this procedure to remove information when estimating the task's true probability, the inputs which are seen by our models remain unchanged. As this removes all word order information when estimating the MLM task's probability distribution, our estimate is only dependent on information about each token's number of appearances in each input.

3.4 Is position information necessary for the task ?

Given the true probabilities p_o and p_u for our task, we want to measure how different these are. We compute the KL-divergence :

$$D_{KL}(p_o, p_u) = \sum_{x,y \in X_o \times Y_o} p_o(y|x) \log \frac{p_o(y|x)}{p_u(y|x)} \tag{1}$$

This statistical distance allows us to estimate how different are the two distributions. We predict that by masking more tokens, the task would increasingly require position information and the divergence would also increase.[6]

[3] i.e. sequences that have no non-terminal label.

[4] We generate sentences along with their true sentence probability in our artificial language until we reach a probability sum superior to 0.75

[5] Note that when using natural languages, automatic collection of sentences in real corpora does not allow access to all possible completions in context, in addition to only providing sparse, and often biased, samples of sentences. Thus the true probability remains unknown, as noted in §3.2.

[6] Note that while the KL-divergence is asymmetric, in this order the quantity represents the information gain achieved by having access to position information.

3.5 Is position encoding useful to the model?

We test two variants of the BERT architecture (Devlin et al., 2019), using Huggingface's Transformer library (Wolf et al., 2020). In the first model, position information is encoded using learned absolute position embeddings,[7] while such explicit encoding is removed from the second. We call such models **BERT** and **NP**. Their hyperparameters are described in App. B. For each model, we compare its probability estimates q in context to the task's true distribution assuming both that position information is present in contexts p_o, and absent p_u. We do so by computing the KL-divergence between q and $p \in (p_o, p_u)$ as follows:

$$D_{KL}(p, q) = H(p, q) - H(p)$$

We estimate the true entropy $H(p)$ for the masked language modeling (MLM) task using either p_o or p_u on our set of generated sentences:

$$\begin{aligned} H(Y|X) &= -\sum_{x,y \in X \times Y} p(x, y) \log \frac{p(x, y)}{p(x)} \\ &= -\sum_{x,y \in X \times Y} p(y|x)p(x) \log p(y|x) \end{aligned} \tag{2}$$

For each context, we compute the true entropy of its completions :

$$\forall x \in X, h_Y(x) = -\sum_{y \in Y} p(y|x) \log p(y|x)$$

And we finally compute the task entropy by averaging these context entropies over our kept masked contexts X_o or X_u :

$$H(Y|X) = \sum_{x \in X} p(x) h_Y(x)$$

We obtain two true task entropy estimates, $H(p_o)$ for ordered contexts, and $H(p_u)$ for unordered ones. For each model, we then estimate the cross entropy to each true distribution. Denoting the model's output probability q, the cross-entropy writes as follows :

$$H(p, q) = -\sum_{x,y \in X \times Y} p(y|x) \log q(y|x)$$

[7] This encoding scheme is widespread in transformer-based models, see Dufter et al. (2021) for an overview

We then use the tasks's true entropy and the model's cross-entropy to compute the KL-divergence. For each model, by comparing $D_{KL}(p_o, q)$ to $D_{KL}(p_u, q)$, we can assess whether the model's estimates fit better the task's probability for ordered contexts, or unordered contexts. If explicit position encoding is necessary, we predict that $D_{KL}(p_u, q)$ should be greater than $D_{KL}(p_o, q)$ for **BERT**, and lower for **NP**. Otherwise, both models should have similar behavior.

3.6 Testing the effect of masking

In this study, we compare **BERT** and **NP** under different amounts of masking. We surmise that increasing that parameter should increase the necessity of using position information, as measured by eq. (1). If this is the case, varying this parameter will allow us to investigate whether position encoding is necessary as the task increasingly requires using that information.

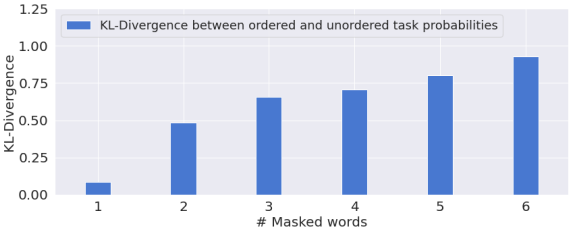

Figure 2: KL-Divergence between the true task probabilities assuming ordered and unordered inputs.

4 Results

We first display the KL-divergence between true probability distributions assuming ordered and unordered contexts in Fig. 2. In accordance with our expectations,[8] when increasing the amount of masking, the true distribution of completions given ordered contexts diverges from that of unordered contexts. Interestingly though, when only one token is masked, the divergence is low. This suggests that in this setting, models should have little difference regardless of whether they have access to explicit position information. By increasing the amount of masked tokens, we can further observe that the two considered true probabilities p_o and p_u diverge. We thus expect that models should increasingly rely on position information to approximate the true ordered distribution.

We further display how well each model approximates each probability estimate in Fig. 3 to verify

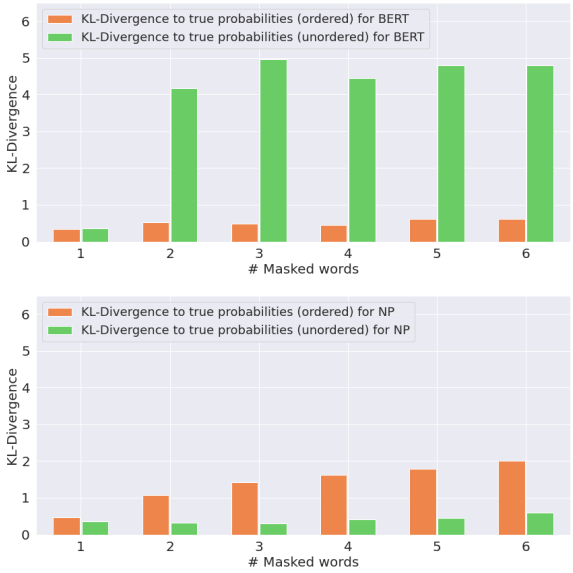

Figure 3: KL-Divergence between the true task probabilities and our models' probability estimates (**BERT**-top and **NP**-bottom), assuming contexts are ordered (orange bars) and unordered (green bars).

whether the presence of position encoding is useful to the masked language modeling task under different amounts of masking. Expectedly, the model with no position encoding scheme performs similarly to the BERT model when only one token is masked. In this setting, the context contains enough information for the model regardless of whether it sees its input tokens as ordered or as a bag of words. When masking more tokens however, this difference becomes increasingly marked.[9]

Further, we observe that the **BERT** model has a low divergence to the true probability assuming ordered contexts regardless of the amount of masking, while it diverges increasingly from the distribution that assumes no ordering of the context. The opposite pattern holds for the **NP** model. Taken together, these results show that position encoding is necessary to approximate the true distribution of the task when it requires position information, that is when the number of masked tokens is increased.

In Fig. 4, we compare our models' cross-entropies to the task's true entropies. The figure aggregates the two main observations made in this article, that when the number of masked tokens increases : (i) the true entropy of the data with and without position diverge from each other, and (ii) that position encoding is required to approximate the task's true probability distribution assuming ordered contexts. Accordingly to our previous ob-

[8] see §3.4

[9] See App. D for our models' perplexities.

servations, the **NP** model, which does not have access to the ordering of tokens, has a cross-entropy that fits the true probability distribution's entropy assuming no ordering of the context (red lines). Looking at **BERT**'s cross-entropy, we see that this model, which has access to position information, rather fits the true probability distribution assuming the context is ordered.

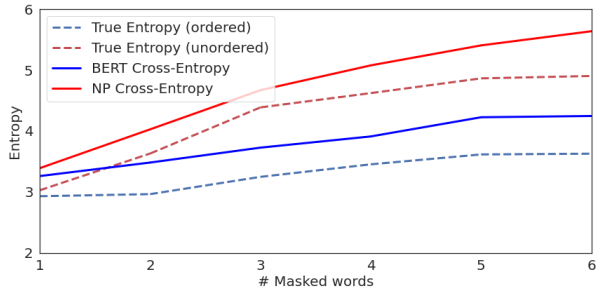

Figure 4: A comparison between entropies of true probabilities for the MLM task (assuming ordered and unordered contexts), and our models' cross-entropies

5 Discussion

5.1 Position encoding and language modeling

Previous work claimed that transformer autoregressive language models without position encodings could reconstruct position information by inferring the number of preceding tokens, but not bidirectional transformer models (Haviv et al., 2022). Testing a RoBERTa model (Liu et al., 2019) led to great difference in perplexity when removing position information at the input level. However, we show this difference to strongly depend on the amount of masking : as autoregressive language models predict only one token at a time, the task could be equally easy for models deprived from position information. Our results call for increased scrutiny when comparing autoregressive and masked language models, making sure that they are asked to predict comparable numbers of tokens.

5.2 Mask more !

In our study, we have shown that the utility of explicit position encoding increases with the number of masked tokens. This finding echoes Wettig et al. (2022)'s study, showing that masking 40% of tokens rather than 15% during pre-training leads to better performance on downstream tasks. This evidence could draw more attention towards understanding how different amounts of masking can lead models to abstract away from position information, and capture more structural knowledge about the languages they model.

6 Conclusion

In this work, we evaluated the importance of position encoding for a masked language model. We showed that without explicit access to position information, a model can obtain performance similar to a model that learns position embeddings, when only one token is masked. We find that when increasing the number of masked tokens, the output probability distribution assuming unordered inputs diverges from that which assumes ordered sentences, reflecting that the task increasingly requires making use of position information. We further show that under this condition, models with position encoding outperform their counterpart deprived from position information. This in turn could draw more attention to the amount of masked tokens, which might be a crucial parameter for models to abstract away from their input sentences' position information, in addition to the chosen position encoding scheme.

7 Limitations

The results we have presented in this paper were obtained over artificial languages. Adapting the method to natural languages may be difficult.

The true probability distribution is not accessible for natural languages. In this study, we investigate how the amount of masking impacts the usage of position encoding by a neural language model. We chose to carry out this experiment on an artificial language, because of the ease to access the true probability distributions in each setting. While this result informs us that the amount of masking could be key for masked language models to use and abstract away from position information extracted from their input, this methodology is not easy to adapt to natural languages, because the true probability distribution is not accessible for natural languages. In future work, one could try to find proxies to estimate reference points for natural languages, with potentialy looser estimates than the one used in this study.

Training several masked language models on natural languages is computationally expensive. In order to investigate how the amount of masking impacts the degree to which a NLM makes use of

its position encodings, or higher-order structural properties of natural languages, one would need to train a large neural model for each condition under investigation, and for each retained amount of masking. This, added to the potential hyperparameter space search would require substantial computing resources as training a model on natural languages requires large amounts of data during training.

Natural languages are usually more flexible regarding word order. In our experiments, we investigate the impact of masking on using position information using artificial languages where word order is fixed. We conclude that neural language models make use of position information on the masked language modeling objective when the number of masked tokens increases. However, while this should hold true for data similar to ours, where the word order is fixed and hence position information greatly affects which token needs to be predicted at a certain position, we cannot make claims regarding the impact of masking on languages where word order is more variable, which is the case of any natural language. Further analyses are needed to evaluate whether position encoding impacts language modeling in different ways when word order is rather fixed (like English), compared to when it is more variable (like in Latin or Finnish).

Ethics Statement

The authors foresee no ethical concerns with the work presented in this paper.

Acknowledgements

This work was funded in part by the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute).

References

Mostafa Abdou, Vinit Ravishankar, Artur Kulmizev, and Anders Søgaard. 2022. Word order does matter and shuffled language models know it. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6907–6919, Dublin, Ireland. Association for Computational Linguistics.

Tyler Chang, Yifan Xu, Weijian Xu, and Zhuowen Tu. 2021. Convolutions and self-attention: Re-interpreting relative positions in pre-trained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4322–4333, Online. Association for Computational Linguistics.

Peng Chen. 2021. PermuteFormer: Efficient relative position encoding for long sequences. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10606–10618, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pu-Chin Chen, Henry Tsai, Srinadh Bhojanapalli, Hyung Won Chung, Yin-Wen Chang, and Chun-Sung Ferng. 2021. A simple and effective positional encoding for transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2974–2988, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Philipp Dufter, Martin Schmitt, and Hinrich Schütze. 2021. Position information in transformers: An overview. *CoRR*, abs/2102.11090.

Adi Haviv, Ori Ram, Ofir Press, Peter Izsak, and Omer Levy. 2022. Transformer language models without positional encodings still learn positional information.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced BERT with disentangled attention. *CoRR*, abs/2006.03654.

Jack Hessel and Alexandra Schofield. 2021. How effective is BERT without word ordering? implications for language understanding and data privacy. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 204–211, Online. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Thang Pham, Trung Bui, Long Mai, and Anh Nguyen. 2021. Out of order: How important is the sequential order of words in a sentence in natural language understanding tasks? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1145–1160, Online. Association for Computational Linguistics.

Ofir Press, Noah A. Smith, and Mike Lewis. 2021. Train short, test long: Attention with linear biases enables input length extrapolation. *CoRR*, abs/2108.12409.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2888–2913, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *CoRR*, abs/2104.09864.

Dingquan Wang and Jason Eisner. 2016. The galactic dependencies treebanks: Getting more data by synthesizing new languages. *Transactions of the Association for Computational Linguistics*, 4:491–505.

Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen. 2022. Should you mask 15% in masked language modeling?

Jennifer C. White and Ryan Cotterell. 2021. Examining the inductive bias of neural language models with artificial languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 454–463, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

A Data Statistics

We describe in Table 1 some statistics for the dataset used to train our models.

Train size	100000
Test & Validation Size	10000
Vocabulary Size	1261
Mean Sentence Length	12.51

Table 1: Statistics of the dataset used to train our models.

B Model Hyperparameters

The architectures' hyperparameters are common to both our **BERT** and **NP** models. The learned tokenizer has been trained without slicing tokens, thus our model's vocabulary is exactly our artificial language's vocabulary.

Layers	3
Attention Heads	4
Hidden Size	256
Intermediate Size	1024
Training steps	300000

Table 2: Hyperparameters of our tested models.

C Training Details

Here we display the parameters used to train our models.

Weight Decay	0.01
Learning Rate	5e-5
Batch Size	8
Optimizer	Adam

Table 3: Hyperparameters used to train our models.

D Model Perplexities

We display the perplexities reached by our models on our validation sets in Table 4. Note that these perplexities are obtained in the traditional masked language modelling setting, where only one word is considered to be the ground truth. This explains the discrepancy when compared to model cross-entropies in Fig. 4. Contrarily to the rest of our analysis, these perplexity scores do not take the true probability distribution of the task into account, as only one label gets all the probability mass.

Model	**# Masked Words**					
	1	**2**	**3**	**4**	**5**	**6**
BERT	15.12	17.06	17.6	19.02	20.24	20.37
NP	20.14	41.93	55.45	70.64	93.58	107.46

Table 4: Perplexities reached by our tested models for varying numbers of masked words.

E Asymmetries in KL-divergences

In Fig. 3, when increasing the amount of masking, the increase in $D_{KL}(p_u, q_{BERT})$ is greater than that of $D_{KL}(p_o, q_{NP})$. While this could look surprising at first glance, it could simply be due to the asymmetric nature of cross-entropy, driven by the non-nullity of its left argument. **BERT** is strongly penalized by the true probability of completions it has never seen in ordered contexts: for those, it should have close to zero probability. **NP** in turn is less penalized, because it should have a non-zero probability for any completion assuming ordered context, as these sets of words are possible completions assuming unordered contexts.